

COS126 Exam 1 Mini-Test

1. Short Answer

1. Here is a 16-bit two's complement binary integer: 111111111101100. Convert it to decimal. Circle your answer.

2. Write the value of `(double) (22 / 7)`.

3. Write the value of `b` after the following two statements are executed. Remember that Java ints use 32-bit 2's-complement representation:

```
int a = 2147483647; // 231 - 1
int b = a + 1;
```

4. Write this number using Java's scientific notation, (without using `Math.pow()`):
 $6.022 \cdot 10^{23}$

5. Write a TOY statement to clear Register 5 to zero.

6. You have a program called `Recipe.java` which reads from standard input and writes to standard output. You have compiled it. The command-line to run it so it reads keyboard input and writes to the terminal screen is: `java Recipe`.

Write the command-line to run it so that, instead of reading input from the keyboard, it reads input redirected from a file named `cookbook.txt`.

Write the command-line to run it so it reads input from `cookbook.txt` and writes to an output file named `meal.txt`

Write the command-line to run it so it reads keyboard input and pipes the output to another compiled program named `HungryThing.java`.

2. Arrays, Functions

The following two methods do the same job. They each take an ORDERED array of ints and a target number as arguments.

```
public static boolean mystery1(int[] array, int target) {
    for (int i = 0; i < array.length; i++) {
        if (array[i] == target) return true;
        else if (array[i] > target) return false;
    }
    return false;
}
```

```
public static boolean mystery2(int[] array, int target) {
    int low = 0;
    int high = array.length - 1;
    while (low <= high) {
        int mid = (low + high) / 2;
        if (array[mid] == target) return true;
        else if (array[mid] < target) low = mid + 1;
        else high = mid - 1;
    }
    return false;
}
```

Use this array to answer the following questions:

```
int[] a = { 2, 5, 11, 14, 15, 27, 31};
```

a) What does `mystery1(a, 5)` return?

b) Fill in the trace table to show that `mystery2(a, 5)` returns the same thing.

target	low	hi	mid	return value

2. Cont'd

c) What do `mystery1(a, 20)` and `mystery2(a, 20)` return?

d) In general, what do these methods do?

3. **Performance.** The following table gives approximate running times for a program with N inputs for various values of N .

<code>N</code>	<code>time</code>
1000	5 seconds
2000	20 seconds
5000	2 minutes
10000	8 minutes

Which of the following best describes the likely running time of this program for $N = 100,000$?

- V. A few minutes
- W. A few hours
- X. Half a day
- Z. A few days

4. Recursion, Debugging

There's a bug in the following recursive program. You need to find it and fix it.

```
public class Series{
    public static int func(int j){
        if (j==1) return 1;
        return 2*func(j-1)+5*func(j-2);
    }

    public static void main(String[] args) {
        int N=Integer.parseInt(args[0]);
        if (N<0) {
            System.out.println("invalid argument");
            return;
        }
        System.out.println(func(N));
    }
}
```

- a. Draw the recursion tree for `func(3)`. You only need to draw the tree up to 3 levels, which means the height of the recursion tree should be no greater than 3.

- b. From the recursion tree in (a), do you see a problem with the program? Explain what is the problem.